IN IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Art Unit	: 1631	**Customer No.: 035811**
Examiner	: Karlheinz R. Skowronek	
Serial No.	: 10/734,023	
Filed	: December 11, 2003	
Inventors	: Anne Vanet	Docket No.: 1421-03
	: Michaela Muller-Trutwin	
	: Thomas Valère	Confirmation No.: 2355
Title	: METHOD FOR IDENTIFYING MOTIFS AND/ : OR COMBINATIONS OF MOTIFS HAVING A : BOOLEAN STATE OF PREDETERMINED : MUTATION IN A SET OF SEQUENCES AND : ITS APPLICATIONS	

DECLARATION OF SOPHIE BROUILLET UNDER 37 C.F.R. 1.132

Mail Stop RCE
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

I hereby declare as follows:

1. I, Sophie BROULLET, am a Bioinformatics Engineer at the Pierre & Marie Curie University Paris 6 (UPMC Paris VI). A copy of my curriculum vitae is attached hereto as Exhibit A.

2. I have read and understood U.S. Patent Application No. 10/734,023 (the "Application"), and have read the Official Action concerning the Application mailed on February 19, 2008.

3. The February 19, 2008 Official Action states that method Claims 1-10, 20, 28, and 29 in the Application are not enabled because they contain subject matter that is not described in the Application in such a way as to enable one skilled in the art to make and use the claimed methods without undue experimentation. I consider one of ordinary skill in the art to be

a person with the equivalent of a Masters degree in Applied Information Technology, Bioinformatics or a related discipline such as Computer Science or Mathematics that has three to four years of work experience relevant to conducting computer based biological sequence analyses.

4. Concerning the alleged lack of enablement of the claimed subject matter in the Application, it is my understanding that one skilled in the art in view of the disclosure in the Application would be able to make and use the claimed methods without undue experimentation.

5. For example, paragraphs [0037] to [0074] and working example paragraphs [0117] to [0130] of the Application provide detailed guidance for performing the claimed methods and constructing the matrices used in these methods to identify motifs which did not mutate together in the collection of sequences (*e.g.* the multiple sequence alignment matrix, NONMUTATED MATRIX B) and to identify couples of motifs which either mutate together or do not mutate together in the collection of sequences (*e.g.* MUTATED MATRIX C).

6. Furthermore, the steps for performing these methods and the rules for constructing the various individual matrices used in these methods (*e.g.* MUTATION MATRIX A, NONMUTATED MATRIX B, and MUTATED MATRIX C) are readily apparent to one of ordinary skill in the art from an inspection of the data values in the various matrices shown in working example paragraphs [0117] and [0130] of the Application. In other words, one of ordinary skill in the art would readily recognize that the operations described below are performed in the working example and the claimed methods. I discuss the working example and these operations below.

7. First, in the working example a multiple sequence alignment matrix is generated. To do this a multiple sequence alignment is performed on a collection of sequences comprising a chain of individual motifs (*e.g.* individual amino acid or nucleotide residues) by using an art recognized method such as the CLUSTALW algorithm. Next, a reference sequence is identified. In the working example the reference sequence is a consensus sequence.

The resulting multiple aligned sequences and consensus sequence form the multiple sequence alignment matrix. In this alignment matrix an aligned sequence can be designated "j" and will have several individual "motifs" (*e.g.* amino acid positions) designated "i" or "k" (*e.g.* amino acid position "zero," "one" *etc.*). In the example the terms "motif" and "position" are used interchangeably. This means that in the example vertical columns labeled "POSITION" are identifying each individual "motif" in the multiply aligned sequences.

The number of sequences in the collection of aligned sequences (SEQ ID NO: 2 to SEQ ID NO: 9), not counting the consensus sequence (*i.e.* the reference sequence SEQ ID NO: 1), is designated "N." In the multiple sequence alignment matrix in the working example the horizontal rows on the left identify each individual sequence, excluding the consensus sequence, and it is apparent the total number of these sequences is "8" so that N = 8.

The number of "motifs" in the aligned sequences are designated "M." In the multiple sequence alignment matrix in the working example the vertical columns labeled "POSITION" identify each individual "motif" and it is apparent that the total number of these "motifs" is "10" so that M=10.

8. Second, the multiple sequence alignment matrix is used to generate MUTATION MATRIX A. This is done by making a matrix with dimensions N x M. This means the number of horizontal rows in MUTATION MATRIX A is the same as "N" (N = 8 in the example) which is the number of sequences in the collection of aligned sequences. This also means the number of vertical columns in MUTATION MATRIX A is the same as "M" (M = 10 in the example) which is the number of motifs (*e.g.* amino acid positions) in the aligned sequences.

The values for each cell in MUTATION MATRIX A are then generated. This is done by comparing a motif in one aligned sequence to the corresponding motif in the consensus sequence. For example, for the cell with coordinates $A_{i,j} = A_{\text{motif identifier, sequence identifier}} = A_{\text{motif 0, SEQ ID NO: 2}}$ a comparison of motif 0 in SEQ ID NO: 2 with motif 0 in the consensus sequence (SEQ ID NO: 1) is made. Then the question: "Is the motif in the aligned sequence the same as the corresponding motif in the consensus sequence?" is asked. If the answer is YES the motifs are the same (*i.e.* the aligned sequence SEQ ID NO: 2 and the consensus sequence SEQ ID NO: 1 have the same amino acid residues at that motif location) then the value entered in that cell location is "1" (which equals "A2" in the example). If the answer is NO the motifs are mutated relative to each other then the value entered in that cell location is "0" (which equals "A1" in the example). Consequently, for the cell in MUTATION MATRIX A with coordinates $A_{i,j} = A_{\text{motif identifier, sequence identifier}} = A_{\text{motif 0, SEQ ID NO: 2}}$ the value entered in the cell is "1" (which equals "A2" in the example) because motif 0 of aligned SEQ ID NO: 2 and motif 0 of the consensus sequence SEQ ID NO: 1 have the same "S" amino acid residue at motif 0.

Similarly, for the cell in MUTATION MATRIX A with coordinates $A_{i,j} = A_{\text{motif identifier, sequence identifier}} = A_{\text{motif 1, SEQ ID NO: 2}}$ the value entered in the cell is "0" (which equals "A1" in the example) because motif 1 of aligned SEQ ID NO: 2 and motif 1 of the consensus sequence SEQ

ID NO: 1 have different amino acid residues at motif 0 (*e.g.* motif 1 of SEQ ID NO: 2 is a "R" residue, while motif 1 of the consensus sequence SEQ ID NO: 1 is a "V" residue).

The steps described above are then performed for each cell in the matrix to produce MUTATION MATRIX A as shown in the working example.

9. Third, MUTATION MATRIX A is used to generate NONMUTATED MATRIX B. This is done by making a matrix with dimensions M x M. This means the number of horizontal rows in NONMUTATED MATRIX B are the same as "M" which is the number of motifs (*e.g.* amino acid positions) in the aligned sequences. This also means the number of vertical columns in NONMUTATED MATRIX B is the same as "M" which is the number of motifs (*e.g.* amino acid positions) in the aligned sequences.

The values for each cell in NONMUTATED MATRIX B are then generated. This is done by going to the top of the vertical columns corresponding to each motif in MUTATED MATRIX A and making a pair-wise comparison of the value "A" (*i.e.* "0" or "1" in the example) entered in each individual cell in the first "motif" column to the A value in the adjacent cell of the second "motif" column. For example, for the cell with coordinates $B_{i,k} = B_{\text{motif identifier i, motif identifier k}} = B_{\text{motif identifier 0, motif identifier 1}}$ a comparison is made of each cell value in the column labeled motif 0 in MUTATION MATRIX A with the adjacent cell value in the column labeled motif 1 in MUTATION MATRIX A. Then the question: "Did any of these individual couples (pairs) of motifs ever mutate simultaneously? is asked. If the answer is YES an individual adjacent couple of motifs mutated at the same time (*i.e.* at least one pair of adjacent cells in MUTATION MATRIX A both have an "A" value of "0") then the value entered in the cell in NONMUTATED MATRIX B is "0" (which equals "B1" in the example). If the answer is NO

an individual adjacent couple of motifs never mutated at the same time (*i.e.* no pair of adjacent cells in MUTATION MATRIX A both have an "A" value of "0") then the value entered in the cell in NONMUTATED MATRIX B is "1" (which equals "B2" in the example). Consequently, for the cell in MUTATION MATRIX B with coordinates $B_{i,k} = B_{\text{motif identifier i, motif identifier k}} = B_{\text{motif identifier 0, motif identifier 1}}$ the value entered in the cell is "1" (which equals "B2" in the example) because no individual adjacent couple of motifs never mutated at the same time (*i.e.* no pair of adjacent cells in MUTATION MATRIX A both have an "A" value of "0").

Importantly, it is apparent how the cell values in NONMUTATED MATRIX B are generated for "self-pairs." For example, for the cell with coordinates $B_{i,k} = B_{\text{motif identifier i, motif identifier k}} = B_{\text{motif identifier 0, motif identifier 0}}$ the value is "B2" = 1 which is determined by a comparison of the cells in the column labeled motif 0 in MUTATION MATRIX A with itself and performing the operations described above. The value in this cell is "1" because no individual adjacent couple of motifs in this "self-pair" mutated at the same time (*i.e.* no pair of adjacent cells in this "self-pair" from MUTATION MATRIX A both have an "A" value of "0"). In fact, motif 0 never changed/mutated as can be seen from MUTATION MATRIX A.

Similarly, for the cell with coordinates $B_{i,k} = B_{\text{motif identifier i, motif identifier k}} = B_{\text{motif identifier 1, motif identifier 1}}$ the value is "B1" = "0" which is determined by a comparison of the cells in the column labeled motif 1 in MUTATION MATRIX A with itself and performing the operations described above. The value in this cell is "0" because an individual adjacent couple of motifs in this "self-pair" mutated at the same time (*i.e.* at least one pair of adjacent cells in this "self-pair" from MUTATION MATRIX A both have an "A" value of "0"). In other words, this is because motif 1 changed/mutated as can be seen from MUTATION MATRIX A.

Applying these rules and spot checking the values in some selected cells of NONMUTATED MATRIX B clearly confirms how the values in each of these cells are generated. In other words, these rules are inherently clear to one of ordinary skill in the art by simply inspecting the values in NONMUTATED MATRIX B.

10. Fourth, MUTATION MATRIX A is used to generate MUTATED MATRIX C. This is done by again making a matrix with dimensions M x M. This means the number of horizontal rows in MUTATED MATRIX C are the same as "M" which is the number of motifs (*e.g.* amino acid positions) in the aligned sequences. This also means the number of vertical columns in MUTATED MATRIX C is the same as "M" which is the number of motifs (*e.g.* amino acid positions) in the aligned sequences.

The values for each cell in MUTATED MATRIX C are then generated. This is done by going to the top of the vertical columns corresponding to each motif in MUTATED MATRIX A and making a pair-wise comparison of the value "A" (*i.e.* "0" or "1") entered in each individual cell in the first "motif" column to the A value in the adjacent cell of the second "motif" column. For example, for the cell with coordinates $C_{i,k} = C_{\text{motif identifier i, motif identifier k}} = C_{\text{motif identifier 6, motif identifier 8}}$ a comparison is made of each cell in the column labeled motif 6 in MUTATION MATRIX A with the adjacent cell in the column labeled motif 8 in MUTATION MATRIX A. Then the question: "Did every motif that mutated do so at the same time (simultaneously) as the second, different motif in the couple? If the answer is YES every motif that mutated did so at the same time as the second, different motif in the couple (*i.e.* are the values in every position of the two columns being compared identical such that every pair of adjacent cells in MUTATION MATRIX A either both have an "A" value of "0" or both have an "A" value of "1") then the

value entered in the cell in MUTATED MATRIX C is "1" (which equals "C1" in the example). If the answer is NO every motif that mutated did not do so at the same time as the other motif in the couple (*i.e.* the values in every position of the two columns being compared are not identical) then the value entered in the cell in MUTATED MATRIX C is "0" (which equals "C2" in the example). Consequently, for the cell in MUTATED MATRIX C with coordinates $C_{i,k} = C_{\text{motif identifier i, motif identifier k}} = C_{\text{motif identifier 6, motif identifier 8}}$ the value entered in the cell is "1" (which equals "C1" in the example) because every motif that mutated did so at the same time as the second, different motif in the couple (*i.e.* the values in every position of the two different columns being compared were identical).

Importantly, it is apparent how the cell values in NONMUTATED MATRIX B are generated for "self-pairs." For example, for the cell with coordinates $C_{i,k} = C_{\text{motif identifier i, motif identifier i}} = C_{\text{motif identifier 0, motif identifier 0}}$ the value is "0" (which equals "C2" in the example) for two reasons. First, motif 0 never mutated. Second, even if a mutation had occurred at motif 0 it could not have occurred at the same time as a mutation in a second, different motif, because in a "self-pair" there is no second, different motif.

Similarly, for the cell with coordinates $B_{i,k} = B_{\text{motif identifier i, motif identifier k}} = B_{\text{motif identifier 1, motif identifier 1}}$ the value is "0" (which equals "C2" in the example). This is because although motif 1 mutated this mutation did not occur at the same time as a mutation in a second, different motif, because in a "self-pair" there is no second, different motif.

11. In light of the foregoing, I believe that given the description in the Application one of skill in the art would be enabled to make and use the claimed methods without undue experimentation. I also believe the rules and operations for performing the claimed methods are

readily apparent to one of ordinary skill in the art from simply inspecting the values in the matrices of the working example.

12. The undersigned declares that all statements made herein of her own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issuing thereon.

18.07.2008
Date

Sophie Brouillet

CURRICULUM VITÆ

BROUILLET Sophie

née le 26 Février 1961
nationalité française

14 rue du moulin de la vierge
75014 Paris
Tel. Dom.: 01 40 44 62 56
Tel. Trv. : 01 44 27 51 23

mail: sophieb@abi.snv.jussieu.fr

Formation à l'université Pierre et Marie Curie (Paris VI)

1985 DEA Océanologie biologique (mention bien)

1986 DESS Informatique appliquée à la biologie (mention très bien)

Activité professionnelle

Depuis le 1er Janvier 2005: Ingénieur d'étude à Paris VI – Développement d'applications bioinformatiques et de sites web.- Gestion de parc informatique.

1994-2005: Ingénieur d'étude à l'Université Versailles- Saint Quentin (Laboratoire Génome et Informatique) – Développement d'applications bioinformatiques et de sites web.

1992-1994 : Ingénieur d'étude contractuel à l'Université Versailles- Saint Quentin (Laboratoire Génome et Informatique)

1988-1991: Ingénieur d'étude contractuel au CNRS Gif sur Yvette (Centre de Génetique Moléculaire) – Développement d'applications diverses et bases de données.

1986-1987: Technicien informatique IFFA CREDO – (Les Oncins 69210 Saint Germain sur l'Arbresle) – Développement d'un logiciel d'épidémiologie hospitalière.

Compétences informatiques

Langages de programmation : C/C++, SQL (Postgresql) Java , PROLOG, LISP
Internet : Administration serveur Apache, HTML, PHP, Javascript
Outils logiciel : Développement (gcc, make, gdb)
OS : Linux, Windows 98 & XP, Mac OSX, UNIX
Administration système UNIX/LINUX, Mac OSX

Divers

anglais (lu, ecrit, parlé), espagnol (lu), russe (lu)